

# A hybrid approach to estimate the complex motions of clouds in sky images



Zhenzhou Peng<sup>a,b</sup>, Dantong Yu<sup>d,\*</sup>, Dong Huang<sup>c,\*</sup>, John Heiser<sup>b</sup>, Paul Kalb<sup>b</sup>

<sup>a</sup> Department of Electrical and Computer Engineering, Stony Brook University, 100 Nicolls Road, Stony Brook, NY 11790, USA

<sup>b</sup> Brookhaven National Laboratory, 2 Center Street, Upton, NY 11973, USA

<sup>c</sup> NASA/GSFC, Mail Code: 613, Greenbelt, MD 20771, USA

<sup>d</sup> Martin Tuchman School of Management, New Jersey Institute of Technology, Newark, NJ 07102-1982, USA

## ARTICLE INFO

### Article history:

Received 26 January 2016

Received in revised form 22 May 2016

Accepted 1 September 2016

Available online 14 September 2016

### Keywords:

Sky imagery

Cloud motion tracking

Optical flow

## ABSTRACT

Tracking the motion of clouds is essential to forecasting the weather and to predicting the short-term solar energy generation. Existing techniques mainly fall into two categories: variational optical flow, and block matching. In this paper, we summarize recent advances in estimating cloud motion using ground-based sky imagers and quantitatively evaluate state-of-the-art approaches. Then we propose a hybrid tracking framework to incorporate the strength of both block matching and optical flow models. To validate the accuracy of the proposed approach, we introduce a series of synthetic images to simulate the cloud movement and deformation, and thereafter comprehensively compare our hybrid approach with several representative tracking algorithms over both simulated and real images collected from various sites/imagers. The results show that our hybrid approach outperforms state-of-the-art models by reducing at least 30% motion estimation errors compared with the ground-truth motions in most of simulated image sequences. Moreover, our hybrid model demonstrates its superior efficiency in several real cloud image datasets by lowering at least 15% Mean Absolute Error (MAE) between predicted images and ground-truth images.

© 2016 Elsevier Ltd. All rights reserved.

## 1. Introduction

The variability and intermittency of generating solar power generation have become one of the biggest obstacles in integrating solar energy into the grid, while maintaining its stable operation (Hanley et al.). To mitigate the volatility of solar power, various solutions have been proposed, such as backup generators, battery reserves, and power system scheduling and dispatches. However, the effectiveness of these solutions largely depends on the accuracy of forecasting in advance the fluctuation of solar power. This need gives rise to a series of research projects in modeling and predicting solar irradiance. As clouds are the primary cause of the intermittency of solar irradiance and the correlated output of a PV system, predicting the variability of solar irradiance essentially becomes an effort to estimate the movement and properties of clouds, in other words, cloud tracking based on current information, and then to create a solar irradiance model from the predicted

properties of the clouds (Hoff and Perez, 2012; Lave and Kleissl, 2013).

Over the past several decades, scientists have utilized remote sensing technology, for example, geostationary satellite imaging, to model and track clouds, and forecast solar irradiance (Escrig et al., 2013; Stuhlmann et al., 1990; Peng et al., 2013). Given the orbit of satellites, these approaches usually provide mid-term forecasts with a time horizon between half hour to six hours, and a spatial resolution of several kilometers. On the other hand, an ever-growing number of distributed roof-top solar panels and smart storage solutions in smart-grids operate in a much shorter time horizons and a smaller spatial resolution than that offered by satellite imagery, and so engenders a pressing need for short-term solar forecasts that predict minutes- or even seconds-ahead solar availability and variability. A promising technique to satisfy the requirement of tracking clouds and predicting solar activities at such a high resolution is the usage of ground-based sky imaging systems. In contrast to satellite imagery systems, the sky imager can visualize local clouds from the ground level in real time, and provide a high spatial resolution of sub-kilometers. Consequently, its output images are widely used for estimating the localized cloud fraction (Long et al., 2006; Pfister et al., 2003) and for

\* Corresponding authors.

E-mail addresses: [zhenzhou.peng@stonybrook.edu](mailto:zhenzhou.peng@stonybrook.edu) (Z. Peng), [dantong.yu@njit.edu](mailto:dantong.yu@njit.edu) (D. Yu), [dong.huang@nasa.gov](mailto:dong.huang@nasa.gov) (D. Huang), [heiser@bnl.gov](mailto:heiser@bnl.gov) (J. Heiser), [kalb@bnl.gov](mailto:kalb@bnl.gov) (P. Kalb).

analyzing the clouds' characteristics (e.g. opacity) (Shields et al., 1993; Singh and Glennen, 2005). More recently, sky imagers, coupled with various cross-domain techniques in the fields of computer vision and machine learning, offer an effective approach to extract the image-based features of clouds, identify their types, estimate their base height, and determine the vertical cloud layers via the clustering technology (Kazantzidis et al., 2012; Savoy et al., 2015; Nguyen and Kleissl, 2014; Peng et al., 2014). Based on the image features extracted and the results of cloud tracking, ground-based imagers gain a remarkable momentum recently to forecast solar irradiance within the range between half minute and up to thirty minutes (West et al., 2014; Bernecker et al., 2014; Yang et al., 2014), which precisely provides a complementary capability to those satellite-based approaches.

The solar forecast systems based on satellites and/or ground-based sky imagers essentially try to predict solar occlusion effect. The model usually involves identification of sun-occlusion pixels associated with ground measurements and estimation of cloud movement to predict the presence of clouds in near future (Chow et al., 2011). Consequently, having an accurate estimation of the cloud's motion is essential to various prediction models that utilize different imagers and target various forecast horizons (Chow et al., 2011; Huang et al., 2012; Haupt and Kosovic, 2015; Xu et al., 2015; Xu et al., 2015).

### 1.1. The limitations with current approaches

Among the techniques of tracking the motions of clouds, block matching and optical flow (OF) are applied widely to various types of imagery, including ground-based cameras and satellites.

Block matching technique takes a collection of pixels (i.e., a block) as a tracking unit, has the ability to utilize regional information, and thereby, is sufficiently robust to image noise and brightness variations within images. If the underlying motions consist of only translative velocity, and do not involve shearing and scaling, a block-matching approach can faithfully represent the true movements of clouds (Huang et al., 2011). However, the majority of block-matching approaches employ pre-defined blocks with a fixed size and position, i.e., a mesh or grid in an image (Hamill and Nehrkorn, 1993; Chow et al., 2011). Consequently, this type of block-matching approach is sensitive to the block's segmentation and an incorrect segmentation in an image can compromise its accuracy. Because non-rigid clouds have a variety of shapes and positions that invalidate any pre-defined segmentation, the performance of these block-based methods is inconsistent over the streams of images from the camera. Even with the assignment of dynamic blocks based on the cloud mask in a sky image (Peng et al., 2014), the performance of the block matching algorithm still deteriorates when multiple pieces of cloud reside within the same block. Many recent methods adopt a hierarchical block structure to track block-wise motions, and apply post-process filtering and interpolating to the tracking outputs (Huang et al., 2011; Mori and Chang). Nevertheless, with these approaches, only a limited improvement is attained, and it still relies on predefined block segmentation and the tedious tuning of block size and position.

On the other hand, optical flow (OF) addresses the motion tracking problem at a lower level than does block-matching. It enables to extract pixel-wise motion through variational methods that first define an energy-like objective function, assume the constancy of brightness cross images, and obtain the solution via minimizing the objective function. Compared to the block-matching approach, the OF model is flexible and can accurately represent complex 3-dimensional motions, such as rotation and scaling, at the pixel level (Héas et al., 2007; Héas and Mémin, 2008). However, it is sensitive to image noise and the variation in brightness, both of which are quite common in sky images. Another issue is that

for texture-less regions or large-motion objects (e.g. fast-moving clouds in sky imagery), the accuracy of OF may drop significantly due to the lack of information (Brox and Malik, 2011). Even with the smoothness term (Golemati et al., 2012) or a regional mask (Wood-Bradley et al., 2012) that incorporate the surrounding information, current OF approaches still neglect the important features, such as the clouds' distribution and the multiple cloud layers, each of which has its own motion. Due to the multi-resolution image processing requirement and iterative optimization scheme, it usually takes minutes or even longer time for OF approaches to extract motion field even from low-resolution images (Sun et al., 2014). Consequently, latency becomes one of the biggest concerns to an OF method and limits its application in short-term or real-time cloud motion tracking with high-resolution images.

### 1.2. Contributions

To fully address the problem of tracking cloud motions, we introduce existing state-of-the-art techniques of both block matching and optical flow in Section 2, and detail seven classic models that have already been applied to estimate cloud motions in sky imagery in Section 3. With observed advantages and disadvantages of prior approaches, in Section 4, we propose an innovative hybrid method that combines both block matching and optical flow to avoid their individual weaknesses and to mutually enhance each other's performance. In summary, we list our contributions in this paper as follows:

1. *Extracting dominant motion patterns.* To incorporate information of cloud layers into motion estimation, we attain the dominant motion vectors from block-wise motions to eliminate outliers and generate motion layers to constrain the optical flow (Sections 4.1 and 4.2).
2. *Formulating a novel optical flow model.* We devise a new OF energy-like function to ensure that optical flow is consistent with the regional trend of cloud motions and at the same time allows small perturbations to track localized deformation in clouds. Afterward, we apply an iterative approach to efficiently minimize the complex objective, and also introduce customized filters to further refine the resulting motions (Section 4.3).
3. *Evaluating our model on simulated and real datasets.* To validate the effectiveness of the tracking methods, we design a comprehensive simulation framework to incorporate translative cloud motions, shape deformation, and various levels of noises into synthetic sky images; then we evaluate our model under these circumstances with the known ground truth. Furthermore, we apply our model to real images collected from various imaging systems and vet its performance in real-world applications (Section 5).

## 2. Related works

Many motion-tracking techniques have been proposed to detect the motion of objects in terms of the pixel-wise movement across different images. However, only a few of them can be used to track cloud motion because of its non-rigid shape and formation. In general, the cloud motion vectors are important to numerical weather prediction models, and usually are obtained from satellite images (Leese et al., 1970; Cote and Tatnall, 1995; Evans, 2006; Corpetti et al., 2008). With the advent of inexpensive digital cameras and the emerging need for solar forecast with the fine granularity that is beyond the spatio-temporal resolution of satellite images, recent researches focused on using these ground-based cameras to track the very short-term motions of clouds (Wood-Bradley et al., 2012; Huang et al., 2013; Chow et al., 2011). These methods fall into three main categories based on the scale and tracking criterion

of the motions. In this paper, we discuss representative works in each category in terms of their approaches and the adopted optimization methods.

### 2.1. Optical flow motion tracking methods

In the field of computer vision, motion tracking is usually resolved as estimating the Optical Flow (OF), i.e., the pixel-wise distribution of prominent velocities of brightness/texture patterns on an image. In general, an OF method can acquire dense motion vectors at pixel level and have been proved effective in cloud motion detection in satellite images (Corpetti et al., 2008). Following original approaches presented in Horn and Schunck (HS) (Horn and Schunck, 1981) as well as Lucas and Kanade (LK) (Lucas and Kanade, 1981), there is a huge body of literature on optical flow. We only focus on several representative models that can effectively address the cloud tracking problem on sky imagery. The core idea of optical flow is to obtain motion field while enforcing the following gradient constraint equation under the assumption of brightness constancy:

$$0 = u \cdot I_x + v \cdot I_y + I_t \quad (1)$$

where  $I$  denotes pixel intensity values in gray scale.  $\nabla I(\mathbf{x}, t) = (I_x, I_y)$  and  $I_t$  denote the spatial and temporal partial derivatives of  $I$  at time  $t$ .  $\mathbf{w} = (u, v)$  denotes the velocity magnitude of each pixel's motion vector in  $x$  and  $y$  direction respectively. The variational equation is known as an ill-posed problem because it only allows to compute optical flow that is normal to image edges. This type of optical flow tracking is classified as *local* method since it optimizes energy-like function for each pixel individually.

To incorporate the neighborhood pixels, Horn and Schunck (1981) introduced the *global* strategy into the optical flow objective function, denoted as HS method. They formulated the optical flow as an optimization problem assuming the global smoothness of motions. The advantage of HS is that it enables the propagation of information over large distances in the image. For texture-less and uniform regions like clear sky or thin clouds on sky imagery, HS extracts the motion information via neighborhood gradient constraints.

Another type of optical flow methods is to constrain motions and preserves local smoothness by satisfying the gradient constraint among neighboring pixels or in a pre-defined local region (LK; Lucas and Kanade, 1981). For sky imagery, the LK method can discriminate the dominant motion vector within cloud/sky regions and is less sensitive to image noise. In particular, LK is adopted by Wood-Bradley et al. (2012) to track clouds in images captured by a laptop camera. Instead of using brightness intensity or grayscale image, they converted the original to the scale of blue-red difference to emphasize its prominent edges and corners. Subsequently, Wood-Bradley et al. extracted prominent features at the points of interest with a large gradient of intensity for calculating optical flow and then manually removed the noisy features to refine the tracking result.

Many variants of the classic models of LK and HS were developed to improve the quality of optical flow. It is beyond the scope of this paper to review the entire literature, and to search for appropriate techniques for the purpose of cloud tracking. Hence, we focus on several typical models that can effectively address the problem of cloud tracking. Chow et al. (2015) utilized a well-known *global* smoothing method in optical flow to detect the clouds' motions on the basis of a new sky imager system (USI Urquhart et al., 2014). Moreover, they adopted the framework of Black and Anandan (1996) and implemented a robust optical-flow approach, termed BA (short for Black and Anandan). Compared with their previous work on estimating the block-wise

motion (Chow et al., 2011), the BA method is claimed to have some advantages: in particular, the error rate of the forecasted binary cloud maps by the BA method with its generated cloud motion vectors is at least 19% less than that by the block-wise motion. Most optical flow models, including BA, adopted a coarse-to-fine warping scheme for constructing a multi-scale image pyramid and iteratively looping through the layers of images for stable outputs of optical flow. However in practice, the coarse-to-fine heuristic quite often does not perform well, particularly for the large movement of objects between two frames. The resulting optical flow fails to faithfully represent the real motion (Brox and Malik, 2011). As fast-moving clouds are common in our observations, we therefore looked into the large-displacement optical flow (LDOF; Brox and Malik, 2011) and compared it with other classic models.

### 2.2. Block-wise motion tracking methods

A block-matching method tries to search for the best-matching blocks from two consecutive images that maximize the pre-defined criterion of similarity. Typically, the criterion of similarity in block-matching techniques is based on cross-correlation or least-square errors, and helps us to find the disparity vector between the same reference block across two images within a constrained window. Different from the OF methodology that assumes constancy in brightness at the pixel level, block-matching techniques assume that the reference block retains textural constancy over time.

Block-matching methods long have been in existence for cloud tracking in satellite imagery (Leese et al., 1970; Rutledge et al., 1991; Evans, 2006). Most used cross-correlation to find the maximum correspondence of regions/blocks between two consecutive image frames (Hamill and Nehrkorn, 1993). In 2011, Chow et al. (2011) adopted cross-correlation as the matching criterion to detect sparse cloud motion vectors (CMVs) in the images of TSI 440A, dividing the original TSI image into the blocks of identical size. As a TSI generates an image from the reflection of dome-shaped mirror, the regular blocks in a raw image are distorted and not uniform in real-space. Therefore, the image distortion compromises the accuracy of the detected motion vectors, especially around the boundary of an image. To resolve this issue, Huang et al. (2011) and Huang et al. (2013) proposed preprocessing TSI images and transforming the original distorted view to a planar view. Thereafter, they located the best-matching blocks based on the Normalized Cross-Correlation (NCC) value, and utilized a refining threshold to remove the low accuracy matches, i.e., the low NCC value. To mitigate the deformation in cloud and predict the variation in block-wise motion, Huang et al. (2013) investigated a multi-frame motion vector tracking and backtracking and extrapolated the future movement with the detected motion trend. However, these approaches are not computationally efficient and cannot take advantage of the cloud's information. To address these issues, we proposed a new method, termed "cloud-block matching" (CBM), to dynamically determine cloud blocks in sky images (Peng et al., 2015), estimate the motion vectors only in the blocks of cloud pixels, and thereby assure its efficiency and efficacy via an intelligent clustering.

### 2.3. Miscellaneous other approaches

In contrast to those two types of tracking, other tracking approaches may use image registration techniques to attain disparity vectors, viz., object motion, between two consecutive frames. For example, phase correlation is a fast noise-resilient approach that estimates the translative offset between two similar frames or sub-frames in the frequency domain (Stone et al., 2001). The phase-correlation method, as shown in the prior work (Huang

et al., 2011), cannot discriminate multiple movements within the same sub-frame region. More importantly, compared with the block-matching technique, the phase correlation technique incurs a higher error rate and is less accurate when no obvious cloud texture is available in the designated regions.

To better address the dynamics of cloud motion, and track the deformation over time, Bernecker et al. (2012) and Bernecker et al. (2014) proposed using a non-rigid image registration. They used the well-studied diffusion model that is developed by Thirion (1998), based on the optical flow method to detect a combined motion vector field with both global translative motions and local variations obtained from the diffusion model. Many new deformation models, such as the flow and curvature models (Sotiras et al., 2013), follow the Demons' method and are widely used in the medical image registration.

In addition, the Particle Image Velocimetry (PIV) methodology is used to estimate cloud motions as a velocity field. Chu et al. (2013) and Marquez and Coimbra (2013) adopted the **MPIV** software that was developed by Mori and Chang to detect the block-wise cloud velocity field in TSI images. Here, **MPIV** partitions an image into reference blocks and searches for the best matching one based on the correspondence criterion of the Minimum Quadratic Difference (MDQ) or the cross-correlation coefficient within a nearby window. Afterward, **MPIV** applies the post-process steps of filtering and interpolation to smooth out the output velocity vectors. However, **MPIV** does not perform well when an image has discontinuities, such as the artifacts of shadowband and the camera's supporting arm in the TSI. To improve the accuracy and aggregate the sparse motion vectors of **MPIV**, Chu et al. (2013) proposed using a k-means clustering to extract two majority motion clusters to differentiate stationary clouds and/or clear sky from fast-moving clouds.

In 2012, Huang et al. introduced a hybrid method that incorporates the stable local descriptor or local feature in estimating cloud motions (Huang et al., 2012). On the top of the block-wise motion, Huang et al. used the Partially Intensity Invariant Feature Descriptor (PIIFD) (Chen et al., 2010) to adjust the motion vectors so to enhance the robustness to geometric and photometric variations. Similar to **LK** motion tracking, Huang's method can determine sparse motion vectors at those points with a large gradient of intensity and correct erroneous block-wise motions.

### 3. Classic models

In this section, we discuss seven classic models that effectively track cloud motions in sky images, detail their implementation, and explore the appropriate parameter settings guided by state-of-the-art works.

#### 3.1. HS optical flow

The original **HS** formulation, proposed in Horn and Schunck (1981), enforces two constraints: The intensity conservation, and the global smoothness among optical flows, and minimize the following energy function:

$$E(u, v) = \sum_{x,y} (u \cdot I_x + v \cdot I_y + I_t)^2 + \lambda (\|\nabla u\|^2 + \|\nabla v\|^2) \quad (2)$$

where  $u$  and  $v$  are the movement in the  $x$  and  $y$  direction respectively at each position  $(x, y)$ .  $\nabla u$  and  $\nabla v$  are the gradients of  $u$  and  $v$ , and can be approximated by subtracting the value at the point  $(x, y)$  from a weighted average of the values at its neighboring pixels (Horn and Schunck, 1981). The iterative algorithm of **HS** uses the regularization weight,  $\lambda$ , to control global smoothness. To achieve optimal performance, we need to tune the smoothness parameter

$\lambda$  carefully for different applications. Even though the heuristic solution of **HS** long has been deemed inaccurate, it still achieves surprisingly good results with a proper optimization and parameter tuning (Geiger et al., 2012).

#### 3.2. LK optical flow

The **LK** methodology (Lucas and Kanade, 1981) relies only on the local information from a small window surrounding the point of interest. Essentially, it is a *local* approach, and can faithfully extract the motion vectors at the points that have prominent features, such as corner points and edges (Wood-Bradley et al., 2012). The energy-like objective function is defined as follows:

$$E(u, v) = \sum_{x,y} g(x, y) (u \cdot I_x + v \cdot I_y + I_t)^2 \quad (3)$$

where  $g(x, y)$  is the weight function to determine the region wherein optical flow is constrained. The Gaussian function is a common example of  $g(x, y)$  (Fleet and Weiss, 2006). Eq. (3) assumes that all pixels within the region share an identical movement. We used the least-square estimator to resolve the optimization of the objective function. During our experiment, we used the *Matlab* implementation of **LK** that entails a coarse-to-fine framework (Bouquet, 2001). We adopted the same settings following (Wood-Bradley et al., 2012), converted the original sky images to the scale of blue-red difference, and then used a three-level image pyramid to iteratively extract **LK** ( $iter_{\#} = 3$ ).  $g(x, y)$  serves as a mask function and represents a region of  $15 \times 15$  pixels. However, in contrast to the original work in Wood-Bradley et al. (2012), our approach generates a dense flow field for all pixels, instead of only for those positions with prominent features.

#### 3.3. BA optical flow

In 1996, Black and Anandan introduced non-convex robust penalty functions into the objective of optical flow to replace the original square-error term (Black and Anandan, 1996). Their method, denoted as **BA**, has been widely applied for various purposes, such as tracking objects and estimating motions in the field of computer vision (Sun et al., 2014). In contrast to previous optical flow approaches, **BA** formulates the energy terms with a series of non-quadratic robust penalty functions  $\Psi(x^2)$  for both the data term,  $E_{color}$ , and the regularization term for smoothness,  $E_{smooth}$ :

$$E(u, v) = E_{color}(u, v) + \lambda E_{smooth}(u, v) \\ = \sum_{x,y} \Psi_D((u \cdot I_x + v \cdot I_y + I_t)^2) + \lambda \cdot \Psi_S(\|\nabla u\|^2 + \|\nabla v\|^2) \quad (4)$$

where  $\Psi_D$  and  $\Psi_S$  respectively are the selected robust functions for data and the smoothness penalty.  $\Psi_D$  and  $\Psi_S$  often are chosen to be the same, e.g. the Lorentzian penalty  $\Psi(x^2) = \log(1 + \frac{x^2}{2\sigma^2})$  in Sun et al. (2010), or the quadratic penalty term  $\Psi(x^2) = x^2$  in the **HS** method. Because Chow et al. (2015) already demonstrated a good performance in tracking cloud motion by utilizing Eq. (4), we evaluate a well-known implementation of **BA** method developed by Sun et al. (2014) and set  $\lambda = 0.1$  based on the experiments in Chow et al. (2015). We also choose the Lorentzian penalty with  $\sigma = 0.03$  as the robust function  $\Psi_D$  and  $\Psi_S$ . In contrast to the original work of **BA**, this new implementation explores the most recent techniques, including preprocessing, filtering, and interpolating methods. Consequently, it significantly improves the accuracy of motion estimation and eliminates noise in the flow field.



### 3.4. Large Displacement Optical Flow (LDOF)

Existing optical flow methods often have difficulties in handling small but fast-moving objects. To overcome this shortcoming, Brox and Malik introduced a new model to track large displacements (Brox and Malik, 2011). Their method, denoted as **LDOF**, formulates the tracking problem with additional energy terms, and, more importantly, incorporates one extra descriptor that regularizes the objective. The idea is similar to the block-wise pipeline designed by Huang et al. (2012) and Wood-Bradley's **LK** model (Wood-Bradley et al., 2012), each of which generates sparse motion vectors based on local or prominent feature matching. Compared with the pixel-to-pixel match used only in those aforementioned approaches, descriptor matching relies on rich local descriptors, such as SIFT and HOG, and is useful for tracking the motions of structures and objects. Hence, this technique was applied successfully to estimate arbitrarily large displacements between frames (Brox and Malik, 2011) or efficiently correct erroneous motion vectors (Huang et al., 2012). The objective function of **LDOF** that is used to estimate pixel-wise motions consists of the following five energy terms:

$$E(u, v) = E_{color}(u, v) + \lambda E_{smooth}(u, v) + \gamma E_{grad}(u, v) + \beta E_{match}(u, v, u_1, v_1) + E_{desc}(u_1, v_1) \quad (5)$$

Note that different from  $E_{color}$  term in Eq. (4), which adopts a linearized form and thereby favors the estimation of motion field with small displacements,  $E_{color}$  in Eq. (5) involves no linearization so to retain the capability of tracking a large displacement of  $(u, v)$ . This is important in estimating cloud motions because fast cloud movements that appear as large displacements between frames are common in sky imagery.  $E_{color}$  between the first frame  $I_1$  and next one,  $I_2$  is defined as follows:

$$E_{color}(u, v) = \sum_{x,y} \Psi(\|I_2(x + u, y + v) - I_1(x, y)\|^2) \quad (6)$$

$E_{smooth}$  is identical to the second penalty term in Eq. (4) while  $E_{grad}(u, v)$  is the supplementary constraint that assumes the constancy of intensity gradient at corresponding points on two frames:

$$E_{grad}(u, v) = \sum_{x,y} \Psi(\|\nabla I_2(x + u, y + v) - \nabla I_1(x, y)\|^2) \quad (7)$$

The remaining two terms,  $E_{match}$  and  $E_{desc}$ , are the energy terms related to the sparse descriptor matching.  $\gamma$ ,  $\lambda$  and  $\beta$  are the parameters that need to be manually tuned based on the various sky images. To improve the performance of motion tracking, Brox and Malik investigated three different methods of descriptor matching: SIFT, histograms of oriented gradients (HOG) and geometric blur (GB). The results show that the HOG method generates the smallest number of mismatches in various dataset. Following their experiments, we applied **LDOF** with HOG descriptor matching and a Gaussian filter with  $\sigma = 0.8$  for cloud tracking in sky images. Moreover, we optimized the parameter setting of **LDOF** based on the model's performance on a training dataset of sky images. In summary, the combined parameter setting consists of  $\lambda = 30$ ,  $\beta = 300$ ,  $\gamma = 5$ .

### 3.5. Hierarchical block matching (BM)

In 2011, Huang et al. (BM; Huang et al., 2011) proposed a two-layer hierarchy of image blocks on grayscale TSI images to guarantee an appropriate granularity for meaningful block matching, i.e., containing sufficient information for the matching calculation, whilst capturing detailed cloud movements. Accordingly, Huang et al. first preprocessed TSI images, divided them into blocks, denoted as reference or inner blocks that, in general, are small

enough to contain only uniform movements. However, small (inner) blocks usually do not have adequate variation in image intensity, so that any block-matching for them may yield inaccurate results. Thereby, instead of using reference block directly, Huang et al. employed a bigger block (outer block) co-centered with each inner block to search for the best match across images. In addition, to refine the matching results and mitigate the influence of artifacts on TSI image, they undertook post-processing to the matching results and applied two thresholds to discard the blocks that have only all black/invalid pixels and/or low matching score. In this paper, we adopted the **BM** method with the same parameter settings outlined in Huang et al. (2011).

### 3.6. Cloud-block matching (CBM)

To incorporate actual cloud information into block matching, we designed the **CBM** method outlined in our prior work Peng et al. (2015) to dynamically determine and match actual cloud blocks. We applied the Support Vector Machine (SVM Lindeberg, 1993) to generate a cloud classifier to separate cloud and sky pixels on the basis of manually annotated images. It utilizes multiple image features to categorize image pixels into two classes (cloud/sky). Once the cloud mask is extracted from a TSI image, a connected component detection algorithm (Haralock and Shapiro, 1991) is applied to the mask to generate the cloud regions that are separated from sky pixels. We assumed that each individual region is a piece of cloud with uniform motion inside, and drew a minimum bonding rectangle for each region to represent the initial cloud block. Lastly, we applied a split-and-merge scheme to divide large blocks or merge small ones, so to ensure that the output of cloud blocks is suitable for the subsequent block-matching computation. Similar to **BM**, the matching criterion of **CBM** is also based the NCC score. In the actual implementation, we specified the range of output block size between  $10 \times 10$  to  $80 \times 80$ . The initial blocks that are outside the size range will be divided or merged.

### 3.7. Particle Image Velocimetry in Matlab (MPIV)

**MPIV** is a Particle Image Velocimetry (PIV) software package to analyze consecutive frames and obtain velocity vectors (Mori and Chang). It generates motion vectors via a hierarchical search, and more importantly, post-processes the generated motion vectors with various de-noising techniques, such as a median filter, an iterative search, and/or motion interpolations. As a result, **MPIV** is widely adopted for detecting motions in various types of images, in particular, tracking clouds successfully in the images from satellites, or ground-based sky cameras (Marquez et al., 2013; Chu et al., 2013; Marquez and Coimbra, 2013). In this paper, we used it to generate velocity fields of clouds across various sky imagers, and compare the results with other state-of-the-art models. Based on the parameter settings in Marquez and Coimbra (2013) to obtain cloud motions in the TSI images, we chose a  $32 \times 32$  pixels processing window and set the overlapping ratio of the search window as 0.5. In addition, we selected the Minimum Quadratic Difference (MDQ) as the matching criterion and set the recursion mode as two to iterate the entire process twice so to enhance the accuracy of the motion vectors.

## 4. New hybrid model

To address the aforementioned issues of optical flow and block-matching, we propose a new hybrid approach that integrates the block-matching method and the variational OF model, and uses the former method to guide/refine the latter one. This new model

encompasses three main steps: (1) extracting cloud mask, generating cloud blocks via bottom-up merging and detecting block-wise motions, (2) identifying dominant motion patterns from detected motion vectors, and, (3) estimating optical flow using our new formulation and refining based on multiple motion filters.

#### 4.1. Block-wise motion detection

First of all, we need to identify all cloud pixels in images and create a cloud mask for subsequent cloud block detection. Here we implement a cloud pixel classification pipeline for **CBM** to generate cloud mask. This pipeline consists of a Support Vector Machine (SVM) based binary classifier that is implemented with the software package *libsvm* (Chang and Lin, 2011) and the parameters setting:  $S = 0$  (Classification),  $C = 10$  (the influence of slack variables),  $t = 0$  (linear kernel). To comprehensively train the classifier for different imagers, we selected four images representing four distinct cloud conditions, scattered clouds, cloudy, overcast, and clear sky, for each sky imager, manually label all cloud pixels, and extract six features for each cloud/sky pixel: R, G, B, red-blue-ratio, Laplacian of Gaussian, and the standard deviation value around the pixel's neighborhood ( $5 \times 5$  in use). With the input labels for the training images and selected features for each pixel, our SVM classifier recognizes cloud pixels under various cloud/lighting conditions with the accuracy of 96.6% (Peng et al., 2015). Lastly, we apply the classifier to generate a cloud mask for sky images where “1” stands for cloud pixels and “0” represents non-cloud pixels.

To aggregate cloud pixels into reference blocks for motion estimation, we use a well-known image decomposition method based on quad-tree structure (Shusterman and Feder, 1994) to divide the cloud mask into blocks, and subsequently estimate block-wise motion vectors via maximizing the NCC score. Here, a quad-tree is a hierarchical data structure in which each node has exactly four children. It is often used to represent an image: each internal node standing for a block of pixels. Starting from the root node that represents the entire image, a quad-tree is generated by recursively decomposing nodes (image blocks) into four children nodes of equal-sized non-overlapping square blocks, until the criterion of homogeneity is met, or the minimum block size is reached. As a result, the whole image is consequently represented with a top-down tree structure. The quad-tree representation efficiently divides a cloud mask into 2-D *homogeneous* cloud blocks that only contain similar color, texture, or structure, and ease the subsequent block-wise motion tracking process.

#### 4.2. Dominant cloud motion patterns

To suppress image noise and neglect falsely estimated motion vectors, we propose to identify the dominant patterns of cloud motions and use them to refine the entire motion field. First, appropriate image preprocessing is necessary to undistort the TSI images ensure that the majority of cloud motions obtained from undistorted image are simple and translative. Subsequently, we can apply a straightforward clustering (e.g. k-means) to effectively group them, to find dominant patterns, and to remove abnormal ones (Chen et al., 2013; Peng et al., 2015).

However, the clustering approach does not take into consideration the weight of each motion vector, i.e., the size of actual cloud block, and thereby, might miss some dominant vectors. In this paper, we utilize the histogram statistics presented in He and Sun (2012) to extract the  $N$  most frequent cloud motions in a sky image, and use them to correct and refine the results of estimated motions. To accurately determine the significant motion modes for cloud pixels and exclude the small-scale motions caused by image noise or sky pixels, we only consider those *obvious* motion vectors

that have adequate velocity between two consecutive frames. In our experiments, we empirically use one as the threshold of vector length to remove small-scale motion vectors ( $\|u\|^2 + \|v\|^2 \leq 1$ ), thereby, ignore stationary sky pixels or slow-moving clouds in calculating dominant motions.

In our implementation, we first use motion vectors obtained from block matching approaches to produce a *dense motion field* where motion vector is resolved at the finest level, i.e., the pixel level, and each cloud pixel acquires the associated motion vector of the cloud block on which it resides. Thereafter, we create a  $10 \times 102$ -D histogram for the motion vectors of all cloud pixels. This design considers the size of a cloud block, and a large block has many pixels, each of which contributes to the count, viz., the weight of the shared block-wise motion vectors to the histogram. Then we would pick the  $N$  highest bins in this histogram that contain the  $N$  most-frequent motion modes assuming that all motion vectors in the same bin share the same mode, and grouped pixel-wise motion vectors into  $N$  clusters. As the number of cloud layers with differing velocities is usually less than four, in practice we used  $N = 4$  throughout the paper for cloud motion tracking. For each motion cluster, we selected the median motion vector as the dominant motion pattern, and thereby, generated a collection of dominant motion vectors ( $\Omega = \{(u_i, v_i) | i = 1, 2, \dots, N\}$ ). Finally, we refine all non-zero motion vectors by resetting their values to the closest dominant motion vectors:  $(\tilde{u}, \tilde{v}) = \underset{(u_i, v_i)}{\operatorname{argmin}} (\|u - u_i\|^2 + \|v - v_i\|^2)$

where  $(u_i, v_i) \in \Omega$ . Here, we use  $(\tilde{u}, \tilde{v})$  as the reference vector to guide the refinement of optical flow in the next section.

#### 4.3. Context-aware variational model and its refinement

The original OF models lack of the contextual information and are agnostic to the existing domain knowledge, and consequently, they often assign erroneous flow vectors to cloud pixels and are sensitive to noises in images. To cope with those issues and utilize the information of motion layers and cloud distribution, we proposed a new OF model. First, we revise the energy-like objective to assimilate the aforementioned dominant motion patterns, and to use them to create a context for calculating the optical flow at each cloud pixel. However, a motion field obtained only from the straightforward process of energy minimization still suffers those issues. The general practice in many prior works is to apply post-processing techniques, such as median filtering (Wedel et al., 2009) or signal-noise-ratio threshold (Mori and Chang) to mitigate the impact of image noise and remove the outliers. Therefore, we also follow the practice and apply three filters thereafter to further assimilate cloud information, utilize dominant motion patterns, and remove noise in the motion field. Given the facts that one iteration is not sufficient to correct the motion-field and the reference vector still needs to be refined to absorb new information and thereby to closely represent the current motion field, we design an iterative algorithm to generate and optimize the dense flow field and to update the subsequent reference motion vectors. This design recognizes that the vectors detected by the BM and the OF models are actually inter-dependent. For the optimal results, they should be integrated into the same framework to ensure them to mutually enhance each other.

Inspired by the two-stage (assimilation and updating) alternative minimization framework in Sun et al. (2014), we propose a new OF model based on a four-step optimization pipeline that iteratively generates a dense flow field, as shown in Fig. 1. In this four-step framework, we iteratively solved and refined the OF result via introducing the auxiliary motion field  $(\hat{u}, \hat{v})$  to: (1) estimate the flow field  $(u, v)$  so that it is nearly identical to  $(\hat{u}, \hat{v})$ , and (2) calcu-

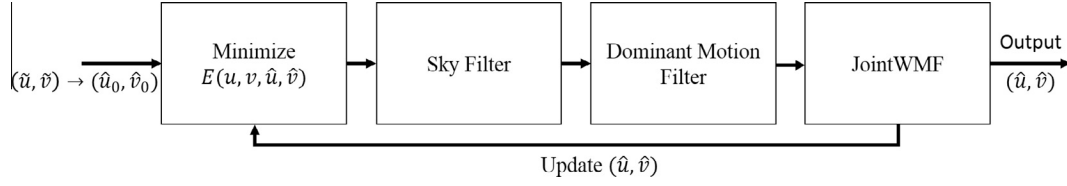


Fig. 1. Minimization of new variational model.

late the best  $(\hat{u}, \hat{v})$  using three motion filters on the current tracking result  $(u, v)$ .

The first step of the pipeline in Fig. 1 is to formulate the OF objective  $E(u, v)$  to be a summation of four energy terms as follows:

$$E(u, v) = E_{\text{color}}(u, v) + \lambda E_{\text{smooth}}(u, v) + \gamma E_{\text{grad}}(u, v) + \beta E_{\text{motion}}(u, v, \hat{u}, \hat{v}) \quad (8)$$

where  $E_{\text{color}}$  and  $E_{\text{grad}}$  are defined in Eqs. (6) and (7).  $E_{\text{smooth}}$  is identical to the second term in BA (Eq. (4)). Here, we add a new energy term  $E_{\text{motion}}$  so to encourage the optical flow to be identical to the reference/auxiliary motion field  $(\hat{u}, \hat{v})$ , whilst allowing a small perturbation from it, and meanwhile penalizing any large deviation.  $E_{\text{motion}}$  is defined as follows:

$$E_{\text{motion}}(u, v, \hat{u}, \hat{v}) = \sum_{x,y} \rho(\|u - \hat{u}\|^2 + \|v - \hat{v}\|^2) \quad (9)$$

The motion perturbation presented in Chen et al. (2013) can track non-rigid transformation and deformation on the local scale. By setting the initial reference motions to be the dominant motion patterns, i.e.,  $(\hat{u}_0, \hat{v}_0) = (\tilde{u}, \tilde{v})$ , for each pixel, we constrain clouds to move along the major trend (wind direction) while permitting a small deviation to capture the local deformation and the changing shape of the clouds.

Optimizing Eq. (8), however, is a non-trivial task since the original problem is not linear. Its minimization requires a two-step linearization based upon the corresponding Euler-Lagrange equations, and successive over-relaxation (SOR) (Brox et al., 2004). The details of discretization and implementation are given in Brox and Malik (2011) and Brox (2005).

However, the motion field obtained by minimizing the energy-like objective still contains anomalies that do not match the real cloud movements. The general practice in many previous efforts was to apply post-processing techniques, e.g. median filters. In the pipeline in Fig. 1, we use three different filters (sky filter, dominant motion filter, and weighted median filter) thereafter for assimilating cloud information, utilizing dominant motion patterns, and de-noising the motion field.

*Sky filter* processes the clear sky pixels that are identified by velocity threshold. We assume that the clear sky regions in images do not move between consecutive frames, and set the motions of all sky pixels that acquire a small-scale motion (i.e.  $u^2 + v^2 \leq T_{\text{sky}}^2$ ) in the previous step to be zero in both the  $x$  and  $y$  directions. It eliminates the small motion vectors caused by image noise and avoids accumulating errors in the iterative minimization step. It is noteworthy that instead of considering all clear sky pixels, we filter out only the ones with a small movement based on the threshold criterion  $T_{\text{sky}}$  because cloud pixels in thin clouds or the images containing a high level of image noise have a high probability of being falsely categorized into the class of clear sky. Consequently, relying on cloud mask only potentially neglects these special cases, and accidentally removes the prominent motions of these mis-classified pixels.

A *dominant motion filter* is designed to identify the outliers with reference to the dominant motion patterns. If a motion vector at

position  $(x, y)$  has significant deviations from all dominant motion patterns  $(\Omega)$ , then we identify it as an outlier, i.e., if a motion vector meets the following condition, we will remove it.

$$\{(u_{x,y}, v_{x,y}) | (u_{x,y} - u_i)^2 + (v_{x,y} - v_i)^2 > \epsilon^2, \forall (u_i, v_i) \in \Omega\} \quad (10)$$

This filter significantly helps to refine the motion field for the next round of minimization and to update the reference motion field.

A *weighted median filter* is widely adopted to smooth and de-noise the motion field in the post-processing step (Sun et al., 2014). Our model employs this filter to update reference (auxiliary) motion field  $(\hat{u}, \hat{v})$ . The weighted median filter can be calculated as follows:

$$\min_{\hat{u}_{x,y}, \hat{v}_{x,y}} \int_{(x',y') \in \Gamma(x,y)} w_{x,y}^{x',y'} (|\hat{u}_{x,y} - u_{x',y'}| + |\hat{v}_{x,y} - v_{x',y'}|) \quad (11)$$

where  $\Gamma(x, y)$  is the set of  $(x, y)$ 's neighborhood pixels  $(x', y')$  within a pre-defined window (e.g.  $5 \times 5$ ) centered at  $(x, y)$ .  $w_{x,y}^{x',y'}$  stands for the weight of the affinity (similarity) between two pixels. In our implementation, we approximate its calculation with the color difference in an image:

$$w_{x,y}^{x',y'} = e^{-\frac{\|F(x,y) - F(x',y')\|^2}{2\sigma^2}} \quad (12)$$

where  $F$  represents the color image with the R, G, B channels. To compute  $w$  efficiently between each pair of pixels, we adopt the joint-histogram weighted median filter (JointWMF Zhang et al., 2014) to reduce the execution time of updating  $(\hat{u}, \hat{v})$ . Compared with the weighted median calculation in Sun et al. (2014), the JointWMF significantly speeds up the running time by more than 10 times and at the same time preserves the quality of estimation (Zhang et al., 2014). Afterward, we pass the updated  $(\hat{u}, \hat{v})$  for the next round of the iterative minimization of  $E(u, v)$ . Based on our empirical experience on sky imagery, our best practice is to iterate the whole process three times to balance between the computational cost and the accuracy of optical flow estimation.

## 5. Experiments

To validate the effectiveness of our proposed model, we propose establishing a comprehensive simulation framework to incorporate translative cloud motions, cloud deformation, and various levels of noises into the synthetic sky images. Then, we can evaluate our model under these circumstances with the known ground truth. Furthermore, we will apply our model to real images collected from various imaging systems and validate its performance in real-world applications.

### 5.1. Simulated dataset of cloud images

To evaluate the robustness of tracking models, we generated three different two-frame sequences of synthetic images from a cloud template (foreground) and a real sky image (background), each of which represents image noises or a different type of cloud motion, i.e., deformation or scaling, as shown in Fig. 2. We first

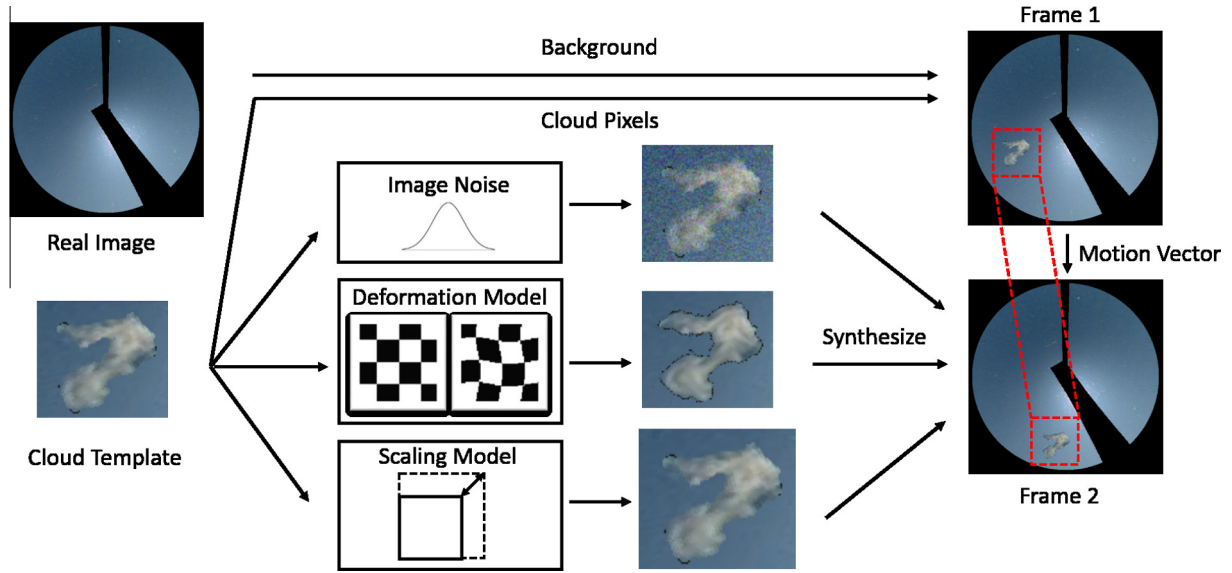


Fig. 2. Framework of synthesizing two-frame sequence of sky images.

overlaid the template on the background image to create the first synthesized frame. Then, we employed different transformation models or injected image noise to generate a new template, and subsequently applied a translative motion vector to synthesize the second frame. Consequently, the ground-truth motion field is a combination of the template translation and the motion perturbations originating from the deformation or scaling models. We detail the three types of simulations as follows:

- (1) *Simulation with a deformation model.* Inspired by prior work (Thirion, 1998) that simulated the changes in the Magnetic Resonance Image of brain (MRI) by a 2-D synthetic deformation model applied on a template, we utilized the same model of image distortion to resemble the changes in shape and deformation of clouds during their movements between two consecutive frames; we also adopted their spatial sinusoidal functions with the same settings to the deformation parameters of amplitude ( $A$ ) and spatial period ( $S$ ).
- (2) *Simulation with scaling.* We utilized an image resizing tool to simulate the cloud's shrinking and expansion. The size of the new template is the scale ratio ( $SR$ ) multiplied by its original size. In our experiment, we set  $SR$  in the range between (0.9–1.2) with a step size equal to 0.1.
- (3) *Simulation with different levels of noises.* Following the idea proposed in Golemati et al. (2012) that added the Gaussian noise to the images of carotid arteries, we also injected the same type of random noise into the R,G,B channels of our synthetic image sequences. In this simulated dataset, the signal-to-noise ratio (SNR) ranged between 20 and 50.

## 5.2. Real sky imagery datasets

In this paper, we use seven types of sky images that were collected from different locations, and/or different types of sky cameras (see Table 1 for details). The column *Rate* stands for the time interval of two consecutive frames of certain image source. The first three datasets, denoted as *BNL1*, *BNL2*, and *BNL3*, were obtained from a small network of three TSIs deployed in the Long Island Solar Farm (LISF) at Brookhaven National Laboratory. These TSIs have an identical configuration, and are placed close to each other so to acquire the overlapping views for the purpose of tracking the micro-scale cloud motions. To cover the representative

conditions of various types of weather and clouds, we chose 562 timestamps from May/07/2013 to June/25/2013 that are synchronized among all three TSI datasets. We converted the raw TSI images ( $480 \times 640$ ) into their planar views to eliminate distortion in the images and cropped the resulting planar images with a resolution of  $500 \times 500$  pixels, and a pre-defined Field of View (FOV) of about  $120^\circ$  to remove low-quality regions. Lastly, we masked out the irrelevant pixels in the areas with artifacts, such as the camera supporting arm, the shadow band, the out-of-FOV areas, and ground obstacles, as illustrated in Fig. 3a and d.

The fourth and fifth dataset are from the TSIs in different locations: *SGPC1* represents the dataset that consists of a pair of two image frames collected daily every 30 min between 10:00 AM and 16:00 PM (local time) from April/15 to June/01/2012 from the central facility of the Southern Great Plains; *TWPC1* is collected from the Tropical Western Pacific site in Manus Island. Similar to *SGPC1*, *TWPC1* comprises the pairs of consecutive frames every 30 min daily from August/25 to November/01/2010. We also pre-processed the raw TSI images of these two datasets with the same method to correct distortions, and the same mask with a FOV of  $120^\circ$ . These two datasets ensured that our proposed model of estimating cloud motions has sufficient coverage for various cloud- and weather-conditions in different geographical locations.

The remaining two datasets were obtained from the high-definition cameras deployed at Brookhaven National Laboratory. The first dataset, *HD-TSI*, is from a high definition TSI in which we replaced the built-in camera by a security camera with a high-resolution of  $2592 \times 1920$  pixels, (Fig. 3b). After preprocessing, we generated a planar view of  $1000 \times 1000$ , with a FOV of about  $140^\circ$  (Fig. 3e). The other HD-image dataset, *HD-SI*, was obtained from a security camera that faced upward to take pictures of local sky regions directly via its fish eye lens (Fig. 3c). After preprocessing, we also retained the images with a resolution of  $1000 \times 1000$  and the same FOV of about  $140^\circ$  (Fig. 3f). These two datasets helped us to validate whether our model is scalable to different types of cameras and large images.

## 5.3. Quality control in real sky imagery datasets

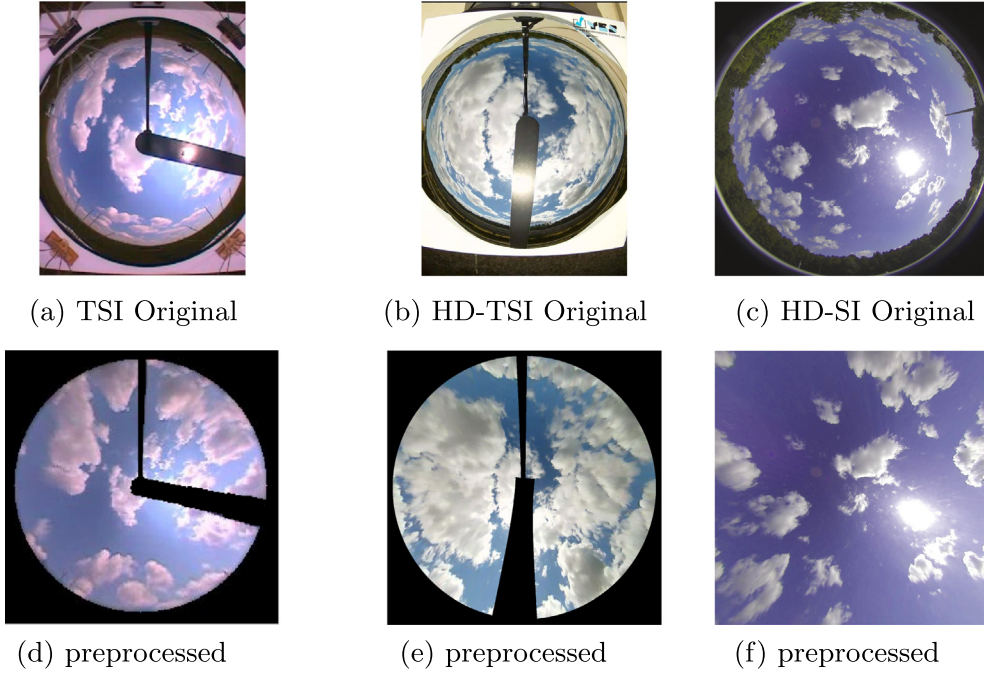
In addition to the preprocessing step, we introduce threshold-based filters to eliminate three types of low-quality images to ensure the veracity of motion tracking. Firstly, images with either



**Table 1**

Overview of seven sky image datasets.

Dataset	Instrument	Range	image <sub>#</sub>	Rate (s)	Res <sub>raw</sub>	Res	FOV (°)
BNL1	TSI 880	2013-05-07–2013-06-25	562	10	480 × 640	500 × 500	120
BNL2	TSI 880	2013-05-07–2013-06-25	562	10	480 × 640	500 × 500	120
BNL3	TSI 880	2013-05-07–2013-06-25	562	10	480 × 640	500 × 500	120
SGPC1	TSI 440A	2012-04-01–2012-06-15	1680	30	480 × 640	500 × 500	120
TWPC1	TSI 440A	2010-08-25–2010-11-01	1654	30	480 × 640	500 × 500	120
HD-SI	IP-cam	2014-10-01–2014-11-01	434	10	1920 × 1920	1000 × 1000	140
HD-TSI	IP-cam	2014-09-01–2014-10-01	434	10	2592 × 1920	1000 × 1000	140

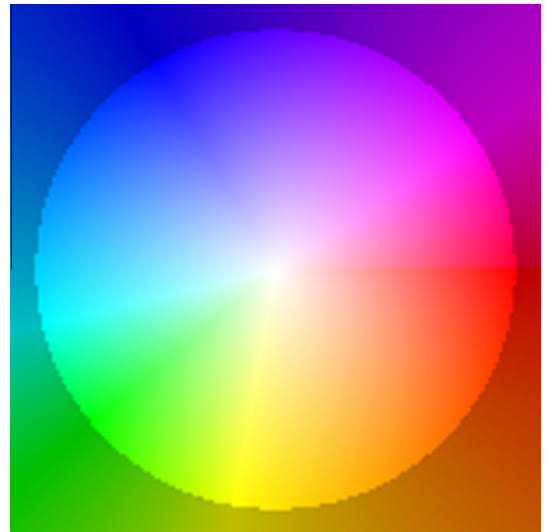
**Fig. 3.** Raw image and preprocessed view of TSI, HD-TSI, and HD-SI.

low solar angles (early in the morning or late in evening) or abnormal color representations, appear to have excessive brightness or darkness, and tend to bear insufficient texture in cloud regions so to generate erroneous motion estimations. For dark images, we filter them out based on their solar angle that is coupled with their capture time. For images with excessive brightness, we apply threshold to select and discard. Furthermore, because we already have three sky imagers with overlapping field of view (FOV), i.e. BNL1, BNL2, and BNL3, we take advantage of the shared FOV and utilize an RGB histogram equalization approach presented in Peng et al. (2015) to correct abnormal color representation in images with defects. Secondly, images that contain only clear sky do not help provide meaningful motion vectors, we filter out clear-sky images based on a cloud fraction threshold ( $<10\%$ ). Thirdly, we discard images with overcast clouds. This type of images usually has texture-less cloud regions, and its cloud movements between consecutive frames are often ambiguous and error-prone for motion estimation because motion tracking models rely on structural difference and horizontal Heterogeneity to best estimate cloud displacement and changes. Therefore, we filter out extremely cloudy or overcast cases based on the cloud fraction threshold  $t_{cf} \geq 90\%$ .

#### 5.4. Evaluation metrics

We adopted five metrics to quantitatively evaluate the performance of our proposed model of estimating cloud motions and to compare it with several representative models. Here, we denote

the estimated motion as  $\mathcal{W}_{(x,y)} = (u, v)$  and the ground-truth motion as  $\mathcal{W}_{x,y}^g = (u_g, v_g)$ . Because a dense motion field is our focus, we will evaluate the average performance of motion tracking on all  $P$  pixels where  $P$  is the total number of pixels in images.

**Fig. 4.** Color coding scheme for the motion vectors in the range between  $(-15, -15)$  to  $(15, 15)$ .

- (1) *Optical-flow color map*. We propose using a color map to visualize a dense motion field, a common approach adopted by computer vision. With it, pixel-level motion vectors are normalized and represented by different colors based on their magnitude and orientation. In this paper we normalize all motion vectors to be within a square ranging between  $(-15, -15)$  to  $(15, 15)$ , and thereafter assign each pixel a color code from the motion color map in Fig. 4.
- (2) *Average angular error (AAE)*. The AAE measures the errors that arise from the angular deviation of an estimated motion vector from the corresponding ground-truth orientation, and is calculated via the following equation:

$$AAE = \frac{1}{P} \sum_{(x,y)} \arccos \left( \frac{\mathcal{W}_{(x,y)} \circ \mathcal{W}_{(x,y)}^G}{\|\mathcal{W}_{(x,y)}\| \|\mathcal{W}_{(x,y)}^G\|} \right) \quad (13)$$

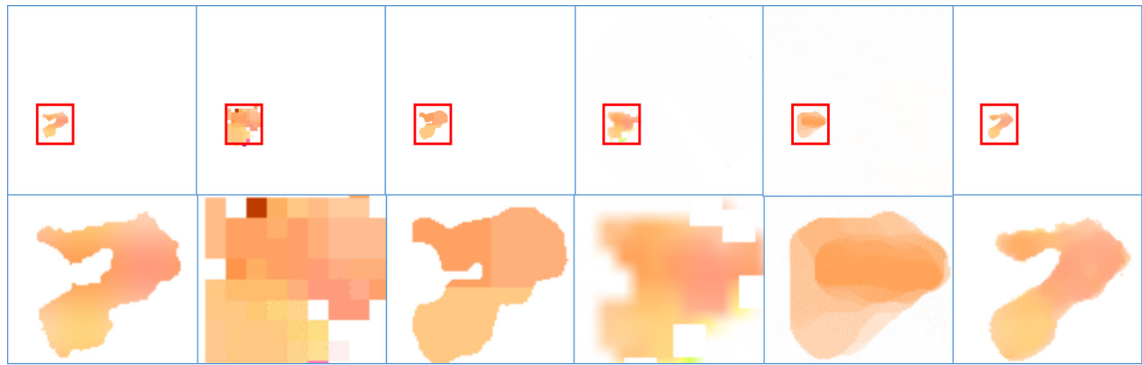
- (3) *Standard deviation of angular error (STDANG)*. STDANG denotes the standard deviation of the angular errors and is defined as follows:

$$STDANG = \sqrt{\frac{1}{P} \sum_{(x,y)} \left( \arccos \left( \frac{\mathcal{W}_{(x,y)} \circ \mathcal{W}_{(x,y)}^G}{\|\mathcal{W}_{(x,y)}\| \|\mathcal{W}_{(x,y)}^G\|} \right) - AAE \right)^2} \quad (14)$$

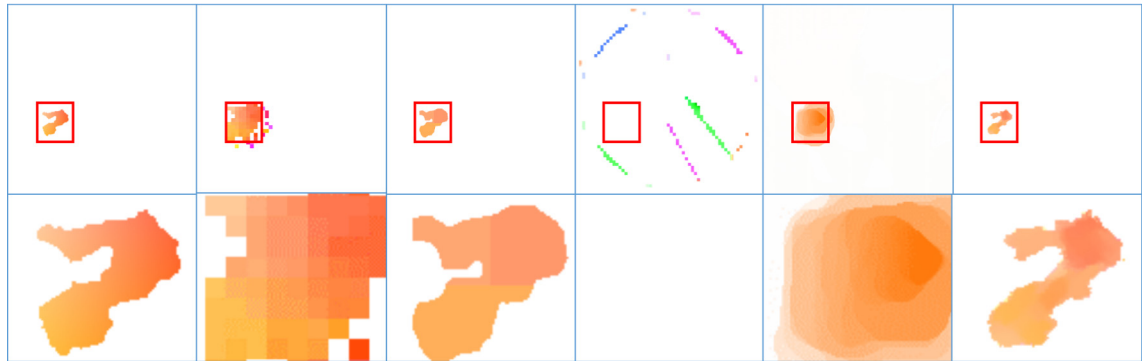
- (4) *Average end-point error (AEPE)*. This evaluates the average deviation of motion vectors from the ground-truth vector.

$$AEPE = \frac{1}{P} \sum_{(x,y)} \|\mathcal{W}_{(x,y)} - \mathcal{W}_{(x,y)}^G\| \quad (15)$$

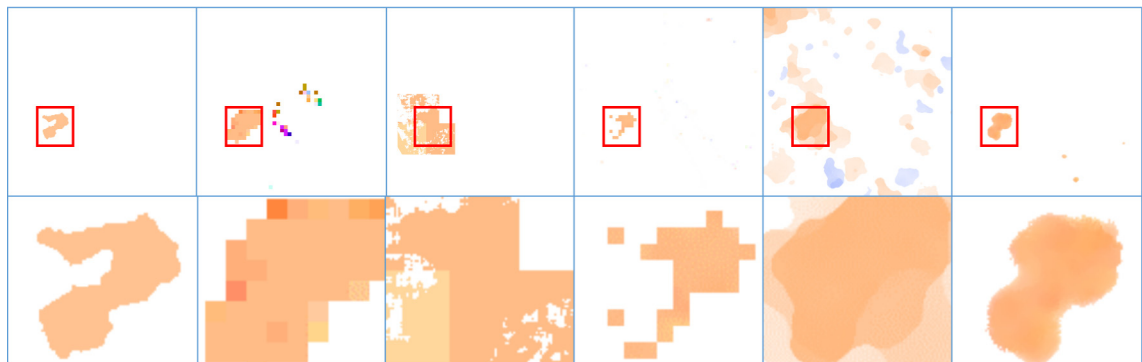
- (5) *Mean absolute error (MAE)*. In our experiment, we use  $I$  to represent the whole image and its grayscale pixel values in the range of  $[0-255]$ . Then the MAE is used to calculate



(a) Deformation ( $A = 2, S = 64$ )



(b) Scaling ( $SR = 1.2$ )



(c) Image Noise ( $SNR = 20$ )

**Fig. 5.** The color map of the results of all tracking models on three types of synthetic images. *Left to right:* Ground-truth motion, **BM**, **CBM**, **MPIV**, **LDOF**, and **Hybrid**.

**Table 2**

The AAE, STDANG, and MEPE scores of the motion-tracking models on simulated images with various deformation settings.

	A = 1, S = 16			A = 3, S = 32			A = 5, S = 32		
	AAE	STDANG	MEPE	AAE	STDANG	MEPE	AAE	STDANG	MEPE
BM	1.771	11.61	0.162	1.692	11.16	0.175	1.705	10.99	0.186
MPIV	0.956	7.710	0.057	0.985	7.829	0.071	1.262	8.934	0.101
CBM	0.368	4.967	0.038	0.501	5.499	0.063	0.660	6.583	0.088
HS	1.573	8.048	0.075	1.551	7.947	0.076	1.572	8.380	0.089
LK	5.277	16.91	0.267	5.456	17.28	0.299	5.761	18.08	0.360
LDOF	0.628	6.671	0.043	0.516	5.393	0.041	0.547	<b>5.401</b>	0.061
BA	0.488	5.587	0.038	0.584	5.500	0.054	0.802	7.061	0.074
Hybrid	<b>0.218</b>	<b>3.852</b>	<b>0.021</b>	<b>0.324</b>	<b>4.435</b>	<b>0.035</b>	<b>0.494</b>	5.655	<b>0.058</b>

**Table 3**

AAE, STDANG, and MEPE scores of motion tracking models on simulated images with three levels of Gaussian noise.

	SNR = 20			SNR = 30			SNR = 40		
	AAE	STDANG	MEPE	AAE	STDANG	MEPE	AAE	STDANG	MEPE
BM	1.813	12.03	0.216	2.355	13.77	0.278	3.271	15.05	0.279
MPIV	0.707	6.582	0.044	0.772	6.427	0.043	1.771	8.710	0.073
CBM	4.097	17.75	0.339	5.003	20.22	1.891	1.034	8.968	0.185
HS	0.637	6.104	0.055	0.715	6.617	0.056	0.799	6.977	0.058
LK	31.17	34.14	1.303	23.95	32.74	1.048	15.51	28.56	0.662
LDOF	33.14	38.04	2.374	2.032	11.95	0.120	1.052	9.026	0.080
BA	16.67	29.28	0.707	6.050	19.33	0.254	1.331	10.09	0.092
Hybrid	<b>0.233</b>	<b>4.225</b>	<b>0.027</b>	<b>0.231</b>	<b>4.224</b>	<b>0.024</b>	<b>0.231</b>	<b>4.241</b>	<b>0.023</b>

the mean absolute error between the ground-truth image ( $I_g$ ), and the predicted one. Here, the predicted image ( $I_{\text{Predict}}$ ) is subsequent to the current image frame in the same sequence and is generated by applying the estimated motion vectors to all pixels thereby mapping those from the current image frame to the next frame. We generally use the MAE score to evaluate the performance of motion estimation (Huang et al., 2012) when the ground-truth for motions does not exist.

$$MAE = \frac{1}{P} \sum_{(x,y)} |I_{\text{Predict}}(x,y) - I_g(x,y)| \quad (16)$$

### 5.5. Experiments on simulated image frames

We applied seven classic models and our proposed hybrid model to estimate cloud motions in the three types of simulated image-sequences. Fig. 5 shows the output motion fields in the format of color map that correspond to the ground-truth, **BA**, **CBM**, **MPIV**, **LDOF**, and **Hybrid** respectively. The first row of images in Fig. 5 indicates the detected motions of all pixels in the entire simulated image whilst the second row is the zoom-in view of the motions of the cloud region (bordered in red). From Fig. 5a and b, we observed that our hybrid model can accurately detect the deformations of clouds and generate the best motion color map, in particular, in the areas around the edges, to closely resemble the ground-truth, when compared with other models. As shown in these two figures, even though **CBM** can estimate the major trend of motions based on its intelligent detection of big blocks of clouds, nevertheless it neglects the underlying deformations in shape within each block, and fails to capture the local variations in the area of clouds. Consequently, its color map appears to be in a solid color with the obvious artifacts arising from the decomposition of a cloud block.

Fig. 5c shows the third case of noises in images. Here, each image is corrupted with the Gaussian noise at a level of  $SNR = 20$ . We observed that because of the existence of these noises, most models fail to recover shape of the cloud and produce

falsely estimated motions for those stationary pixels. With the help of the filters and the dominant patterns of motion, our proposed hybrid model preserves a reasonable segmentation between the motion fields of a cloud region and the area of clear sky, whilst incurring only a much smaller number of outliers in the area of clear sky than do the remaining models.

Tables 2 and 3 show the performance number of all the models with regard to two different deformation settings and multiple levels of noise. We conclude that the **Hybrid** model consistently outperforms other models in most cases in terms of three different metrics, i.e., AAE, STDANG, and MEPE. For the most difficult case of simulating deformation (i.e.  $A = 5$ ,  $S = 32$ ), **Hybrid** retains the lowest level of errors in terms of AAE and MEPE and the second to the lowest in terms of STDANG, which is only 5% higher than the best score obtained by the **LDOF** model.

### 5.6. Experiments on TSI and High Definition (HD) images

To validate the accuracy of detected motion vectors, we applied the resulting motion field between two consecutive frames ( $t - 1$  and  $t$ ) to predict the next frame ( $t + 1$ ) and to compare it with the true image. In Fig. 6, the first row lists the raw data of three frames at time  $t - 1$ ,  $t$ ,  $t + 1$  (ground-truth) and a motion mask to remove outliers and calculate the MAE score. The second row presents the motion fields of **BM**, **MPIV**, **BA**, and **Hybrid** respectively. The third row shows the predicted images using detected motions and image interpolation. The fourth row displays the color maps of the difference between the predicted images by the four models and the ground-truth frame at time  $t + 1$ . We observed that our **Hybrid** model can generate stable motion vectors near the boundary of each piece of cloud. In contrast to other models, **Hybrid** is able to capture the small changes of clouds and produce less MAE errors in texture-rich regions.

In our experiment, we selected eight representative cases to encompass various cloud conditions that appeared in different sites and sky imagers. The first six of them shown in Fig. 7 are in TSI datasets: Fig. 7a shows the condition of scattered clouds whose boundaries are clearly segmented. All pieces of cloud have negligi-



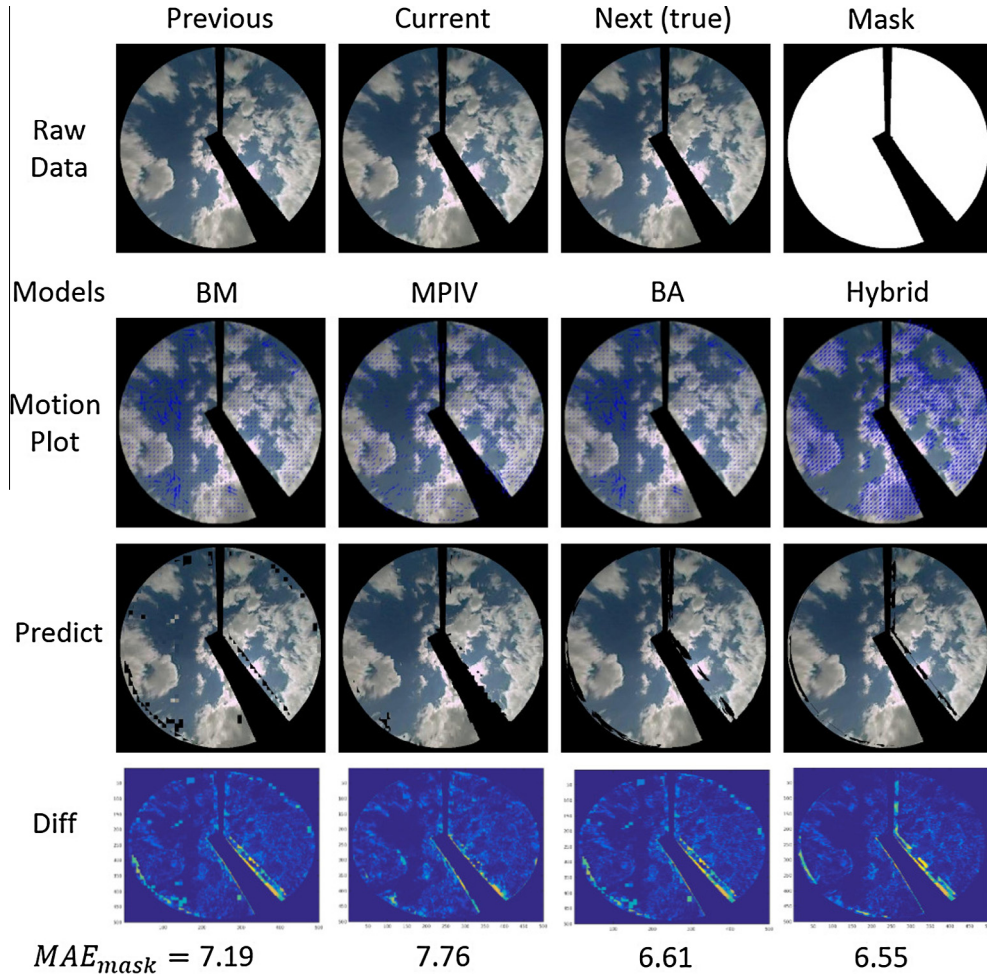


Fig. 6. Using the estimated motion fields to predict the next image frame at time  $t + 1$ .

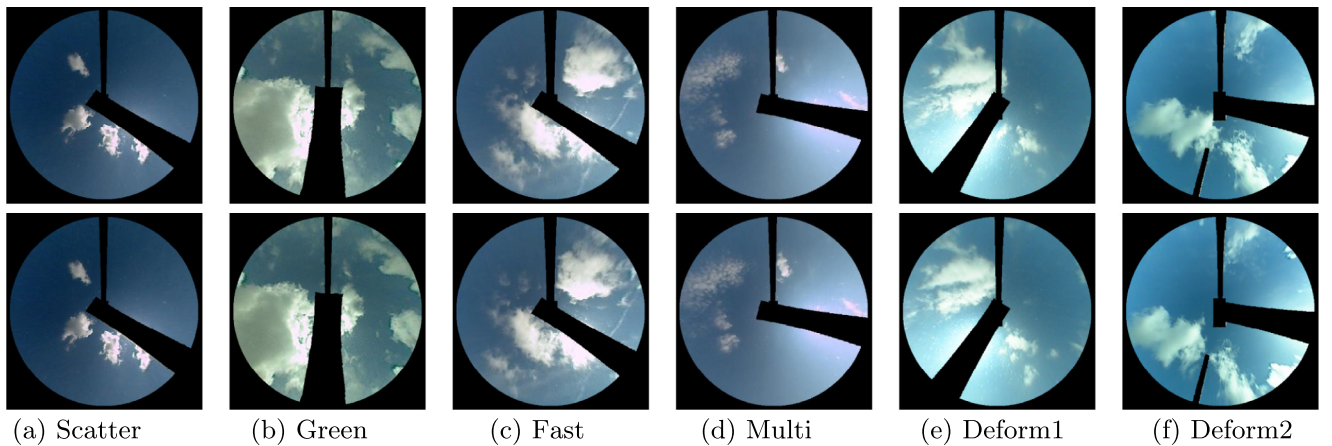


Fig. 7. Six cases selected from the TSI datasets. (a) Scattered clouds; (b) images from *BNL1* dominated by a green color; (c) fast-moving clouds in dataset *BNL3*; (d) multi-layer clouds; (e) and (f) images from two different TSI sites (*SGPC1* and *TWPC1*), both showing fast motion of clouds and deformation in their shape. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ble deformations and identical motion vectors. Fig. 7b shows an abnormal image with a different color scale by an artifact of the camera's setting and dominated by green. Fig. 7c contains fast-moving clouds with a velocity of approximate  $(u, v) = (-17, 2)$ . Fig. 7d comprises two layers of clouds, each with different motion vectors: the first one has  $(u, v) = (6, 1)$  whereas the other has  $(u, v) = (11, -6)$ . This example targets at evaluating each model's

capability to determine the multiple layers of cloud motion. Fig. 7e and f, respectively, come from the dataset of *SGPC1* and *TWPC1*. The clouds in these two examples move quickly between the frames, and more importantly, involve local deformations, such as cloud dissipation and changing shape. The average translative movement in the first example is close to  $(-25, -19)$ , and the average motion during half a minute in the second one is about



(−39, 10). The remaining two cases are from the HD datasets presented in Fig. 3e and f.

Real-world satellite and sky images rarely have detailed ground-truth cloud motions at the level of a pixel or a block of pixels. Given the rarity of detailed ground-truth motion vectors, many approaches rely on both manual procedure of visually checking and compute-aided quantitative analysis and refinement (Escrig et al., 2013; Leese et al., 1971). For example, Leese et al. (1971) tried to determine cloud movements by combining the manual procedure of viewing loop movies with cross-correlation matching for manually identified targets. Inspired by this approach, we design our two-step annotation procedure: manually identify

cloud pieces (cloud regions) in selected images (six cases in Fig. 7), and then estimate translative motion vectors for each region using a cross-correlation based block matching algorithm. Because human eyes provide the best image segmentation to identify different cloud regions, this manual method ensures the optimal performance in the subsequent block matching and is less sensitive to image noise. Therefore, we treat the motion tracking results obtained from manual annotation as the “ground-truth”, and use them to generate a “false” color image map (termed **Manual**) and compare with the results of all motion estimation models, including **BM**, **MPIV**, **BA**, **LDOF**, and **Hybrid** (Fig. 8). Lastly, we apply the resulting motion field to the first image frame so to pre-

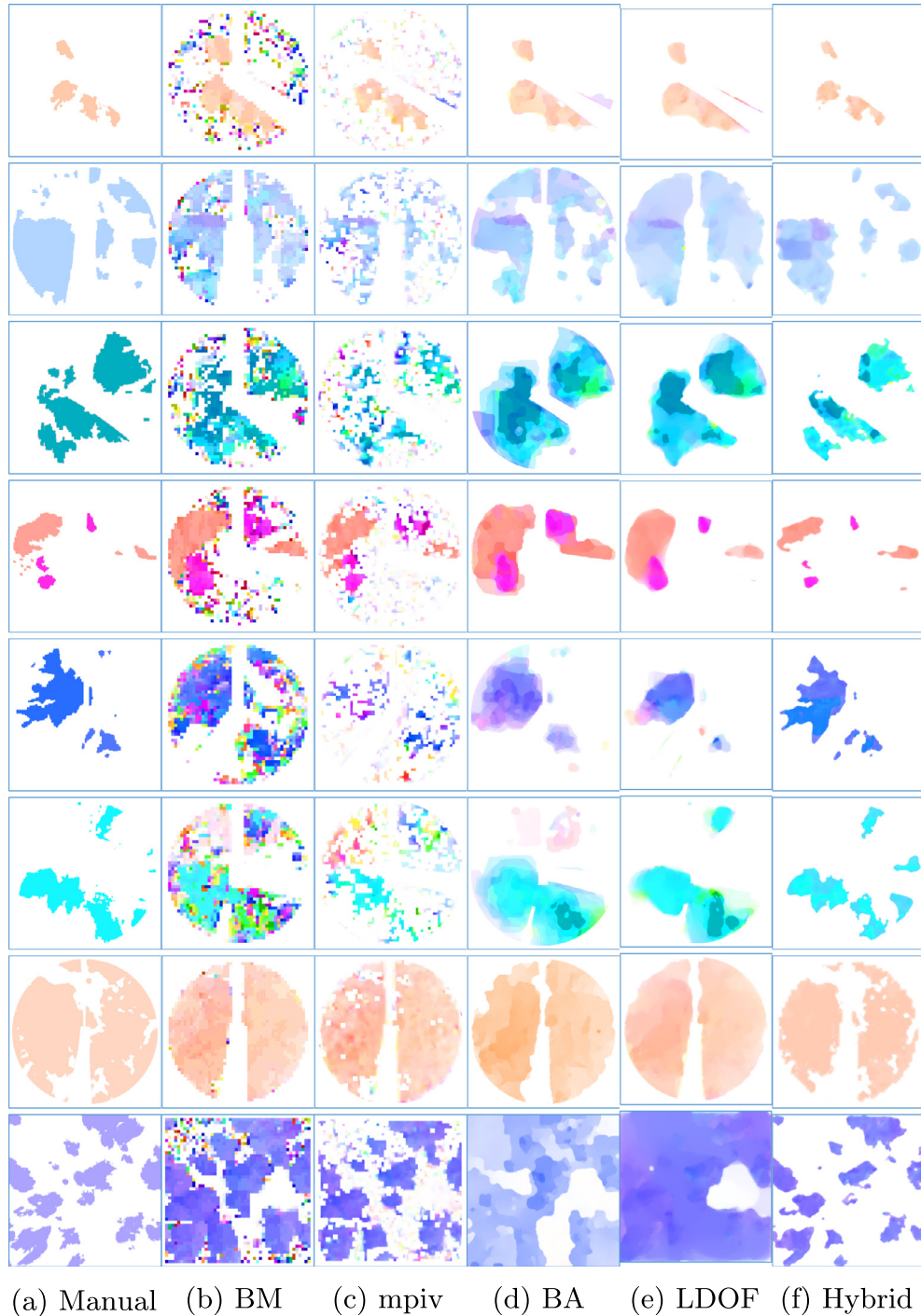


Fig. 8. Color map on eight selected cases.

**Table 4**

STDANG score of eight motion tracking models on simulated and real images. Row *Deform* shows the tracking results of the simulated image pair with deformation model using  $A = 5$ ,  $S = 32$ . Row *BNL3* presents the results of the real images from BNL3 dataset.

	BM	MPIV	CBM	HS	LK	BA	LDOF	Hybrid
Deform	10.99	8.93	6.58	8.38	18.08	5.40	7.06	5.66
BNL3	39.35	21.02	11.42	15.17	28.48	15.83	17.58	9.30

**Table 5**

MAE score of models of eight selected cases.

	Scatter	Green	Fast	Multi	Deform1	Deform2	Fig. 3e	Fig. 3f
BM	19.6	23.5	39.4	33.9	54.9	54.5	13.1	19.6
MPIV	7.8	19.1	29.0	15.6	34.8	44.4	16.9	18.6
BA	3.7	11.4	18.2	6.4	20.3	25.1	12.4	10.7
Manual	3.0	8.0	15.0	4.0	18.0	22.0	8.8	7.3
Hybrid	3.3	8.2	15.4	4.6	18.5	22.5	9.3	8.8

**Table 6**

MAE score of motion tracking models on seven real images.

	BNL1	BNL2	BNL3	TWP	SGP	HD-SI	HD-TSI	Overall
BM	20.4	18.0	24.1	34.6	39.0	18.8	15.7	24.4
MPIV	20.3	15.3	23.6	26.1	27.5	19.6	15.0	21.1
CBM	7.9	5.0	8.7	11.9	15.6	13.4	5.8	9.8
HS	13.5	7.8	11.1	17.8	16.3	18.8	10.9	13.7
LK	42.9	38.4	48.4	51.9	56.1	40.2	39.9	45.4
BA	9.8	6.6	10.9	11.6	11.5	12.5	7.6	10.1
LDOF	7.9	5.0	7.0	11.1	9.8	9.3	5.7	8.0
Hybrid	6.2	4.2	4.7	9.8	8.8	8.7	5.3	6.8

**Table 7**

Running time (in min) of 2-frame cloud motion tracking using four image datasets.

	BM	CBM	MPIV	LK	HS	BA	LDOF	Hybrid
BNL1	0.12	0.02	0.94	0.72	0.11	3.64	1.48	0.17
SGP	0.08	0.01	1.04	0.75	0.12	3.49	1.44	0.22
HD-SI	0.44	0.69	21.2	3.26	0.51	23.8	24.5	0.53
HD-TSI	0.33	0.11	12.1	2.72	0.40	17.8	8.04	0.58

dict the subsequent frame and calculate the MAE score between the predicted frame and the second frame (Table 5). We observed that **Hybrid** attains the best approximation to the manually annotated color map. More importantly, **Hybrid** is robust in all of the eight representative cases and has the fewest MAE errors among the five models.

To quantitatively analyze each model's performance on real sky images, we applied the aforementioned method to manually annotate motion vectors in real images, treated them as ground-truth, and measure tracking errors w.r.t. the ground-truth. Thereby, we can compare the results with the same metrics between simulated and real images. In detail, we generated a pair of synthetic images using the deformation parameters of  $A = 5$ ,  $S = 32$ , and chosen a pair of real images from BNL3 which have the similar color, cloud fraction, and motions to the synthetic image. Table 4 summarizes the STDANG score of eight models for both synthetic and real TSI datasets. Being consistent with its performance in simulation tests, our **Hybrid** model again demonstrates its robustness by obtaining the lowest error rate in TSI images among all models. Meanwhile, we observed that real-image motion estimation is more challenging than one with synthetic images based on the results in Table 4. We attributed this observation to two reasons: (1) real images contain much more noise and texture variations between consecutive frames than synthetic images, resulting in problematic motion vectors, and (2) "ground-truth" in TSI dataset may be biased and deviate from real cloud motions. Even the manual approach generates the optimal image segmentation for enhancing the accuracy of

subsequent cross-correlation matching, it only considers translative motion, and consequently, fails to detect local deformations, such as shape changes. In contrast to our simulation images in which the ground-truth cloud movements are known as a prior, real images do not have prior knowledge and must rely on manual annotation that may not faithfully represent true motion at a pixel level and introduce bias. Consequently, the bias contributes to the estimation error by various tracking models.

Table 6 summarizes the MAE score of all the motion estimation models over different real image datasets. It shows that the **Hybrid** model outperforms the remaining classic models for the TSI and HD imager datasets. Compared with the best-performing block matching approach, **CBM**, and the most accurate OF method, **LDOF**, our proposed hybrid approach further cuts down the overall error rate by more than 30% and 15%, respectively.

### 5.7. Comparison of computation time

To evaluate the efficiency of motion tracking models for short-term solar forecast, we compare the running time of different models using images from both TSI and HD image datasets. Table 7 summarizes the computation time of seven state-of-the-art models, along with our hybrid approach over two low-resolution and two high-definition datasets: BNL1, SGP, HD-SI and HD-TSI. All the models here are implemented in Matlab and tested on a 64-bit Linux laptop with one 4-core CPU of 2.60 GHz and 8 GB memory.

In general, OF methods take more time than block matching for computing the dense motion field. One exception is the **HS** model. In our experiments, its implementation involves no multi-resolution optimization scheme, and thereby is faster but with a trade-off between tracking accuracy and latency. Different from other block matching approaches, **MPIV** has additional costs due to its recurring motion estimation and post-processing filters. From the results of the low-resolution TSI images, i.e. BNL1 and SGP, we observed that all OF approaches are capable of serving forecasts between five and ten minutes ahead. However, high resolution images lead to the dramatic increase to the latency of OF methods, which makes them impractical to support short-term solar forecast.

In contrast, our **Hybrid** model has comparable performance to standard block-based motion algorithms. For HD-SI images, it is even faster than the cloud-based block matching (**CBM**) model. Because the motion tracking process dominates the end-to-end prediction pipeline (Peng, 2016), the processing speed of the hybrid approach is sufficient for predictions between two and fifteen minutes ahead. With new image frame arriving every ten seconds, a single motion tracking pipeline cannot keep up with the arriving rate of cloud image streams. We can apply parallel processing to resolve this problem and avoid dropping frames: in particular, deploying multiple pipelines ( $\geq 4$ ) of the **Hybrid** model into a multi-core server, and dispatching newly arrived frames in a round-robin fashion to an idle pipeline that has its designated CPU core, and generating a new batch of motion vectors every ten seconds.

## 6. Conclusions

In this paper, we discuss our investigation of state-of-the-art motion estimation methods for tracking and predicting motion of clouds, and quantitatively analyze classic models of both block matching and optical flow using various sky imagers and synthetic images. By taking advantage of all prior models, we propose a new hybrid model to employ cloud mask and quad-tree in block-wise motion-tracking, to extract the dominant motion patterns via histogram statistics, and to estimate a dense motion field at pixel level via customized motion filters and objective function.

To validate the effectiveness of our model for cloud motion estimation, we not only collected various types of real images from different sites, but also synthesized three image-sequences to simulate the shape-changing clouds and image corruption with random noise. The results show that compared with classic models, our new model can accurately capture cloud deformation and is resilient to image noise. It consistently produces the best color map and greatly reduces the angular error (AAE) by at least 30% in all simulated image sequences. In terms of real images, the new model consistently extracted clear motions of clouds under various conditions (e.g. multiple layers and fast-moving clouds) and lessened MAE by at least 10% for both TSI and HD image datasets.

## References

- Bernecker, D., Riess, C., Angelopoulou, E., Hornegger, J., 2012. Towards improving solar irradiance forecasts with methods from computer vision. In: *Computer Vision in Applications Workshop*.
- Bernecker, D., Riess, C., Angelopoulou, E., Hornegger, J., 2014. Continuous short-term irradiance forecasts using sky images. *Solar Energy* 110, 303–315.
- Black, M.J., Anandan, P., 1996. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Understand.* 63 (1), 75–104.
- Bouguet, J.-Y., 2001. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporat.* 5, 1–10.
- Brox, T., 2005. From Pixels to Regions: Partial Differential Equations in Image Analysis, Ph.D. thesis.
- Brox, T., Malik, J., 2011. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (3), 500–513.
- Brox, T., Bruhn, A., Papenberg, N., Weickert, J., 2004. High accuracy optical flow estimation based on a theory for warping. In: *Computer Vision-ECCV 2004*. Springer, pp. 25–36.
- Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2, 27:1–27:27. Software available at <<http://www.csie.ntu.edu.tw/~cjlin/libsvm>>.
- Chen, J., Tian, J., Lee, N., Zheng, J., Smith, R., Laine, A.F., 2010. A partial intensity invariant feature descriptor for multimodal retinal image registration. *IEEE Trans. Biomed. Eng.* 57 (7), 1707–1718.
- Chen, Z., Jin, H., Lin, Z., Cohen, S., Wu, Y., 2013. Large displacement optical flow from nearest neighbor fields. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2443–2450.
- Chow, C.W., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Idris, J., Washom, B., 2011. Intra-hour forecasting with a total sky imager at the UC san diego solar energy testbed. *Solar Energy* 85 (11), 2881–2893.
- Chow, C.W., Belongie, S., Kleissl, J., 2015. Cloud motion and stability estimation for intra-hour solar forecasting. *Solar Energy* 115, 645–655.
- Chu, Y., Pedro, H.T., Coimbra, C.F., 2013. Hybrid intra-hour dni forecasts with sky image processing enhanced by stochastic learning. *Solar Energy* 98, 592–603.
- Corpetti, T., Heas, P., Mémén, E., Papadakis, N., 2008. Variational pressure image assimilation for atmospheric motion estimation. *IGARSS 2008. IEEE International Geoscience and Remote Sensing Symposium*, 2008, vol. 2. IEEE, pp. II–505.
- Cote, S., Tatnall, A., 1995. A neural network-based method for tracking features from satellitesensor images. *Int. J. Rem. Sens.* 16 (18), 3695–3701.
- Escrib, H., Batlles, F., Alonso, J., Baena, F., Bosch, J., Salbidegoitia, I., Burgaleta, J., 2013. Cloud detection, classification and motion estimation using geostationary satellite imagery for cloud cover forecast. *Energy* 55, 853–859.
- Evans, A.N., 2006. Cloud motion analysis using multichannel correlation-relaxation labeling. *IEEE Geosci. Rem. Sens. Lett.* 3 (3), 392–396.
- Fleet, D., Weiss, Y., 2006. Optical flow estimation. In: *Handbook of Mathematical Models in Computer Vision*. Springer, pp. 237–257.
- Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The kitti vision benchmark suite. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 3354–3361.
- Golemati, S., Stoitsis, J.S., Gastounioti, A., Dimopoulos, A.C., Koropouli, V., Nikita, K.S., 2012. Comparison of block matching and differential methods for motion analysis of the carotid artery wall from ultrasound images. *IEEE Trans. Inform. Technol. Biomed.* 16 (5), 852–858.
- Hamill, T.M., Nehrkorn, T., 1993. A short-term cloud forecast scheme using cross correlations. *Weather Forecast.* 8 (4), 401–411.
- Hanley, C., et al., Solar Energy Grid Integration Systems-Energy Storage, SAND2008-4247, Sandia National Laboratories, Albuquerque, NM.
- Haralock, R.M., Shapiro, L.G., 1991. *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc.
- Haupt, S.E., Kosovic, B., 2015. Big data and machine learning for applied weather forecasts: forecasting solar power for utility operations. In: *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, pp. 496–501.
- He, K., Sun, J., 2012. Statistics of patch offsets for image completion. In: *Proceedings of the 12th European Conference on Computer Vision, ECCV'12*, vol. Part II. Springer-Verlag, Berlin, Heidelberg, pp. 16–29. [http://dx.doi.org/10.1007/978-3-642-33709-3\\_2](http://dx.doi.org/10.1007/978-3-642-33709-3_2).
- Héas, P., Mémén, E., 2008. Three-dimensional motion estimation of atmospheric layers from image sequences. *IEEE Trans. Geosci. Remote Sens.* 46 (8), 2385–2396.
- Héas, P., Mémén, E., Papadakis, N., Szantai, A., 2007. Layered estimation of atmospheric mesoscale dynamics from satellite imagery. *IEEE Trans. Geosci. Remote Sens.* 45 (12), 4087–4104.
- Hoff, T.E., Perez, R., 2012. Modeling PV fleet output variability. *Solar Energy* 86 (8), 2177–2189.
- Horn, B.K., Schunck, B.G., 1981. Determining optical flow. In: *1981 Technical Symposium East. International Society for Optics and Photonics*, pp. 319–331.
- Huang, H., Yoo, S., Yu, D., Huang, D., Qin, H., 2011. Cloud motion detection for short term solar power prediction. In: *Proceedings of ICML 2011 Workshop on Machine Learning for Global Challenges*.
- Huang, H., Yoo, S., Yu, D., Huang, D., Qin, H., 2012. Correlation and local feature based cloud motion estimation. In: *Proceedings of the Twelfth International Workshop on Multimedia Data Mining*. ACM, pp. 1–9.
- Huang, H., Xu, J., Peng, Z., Yoo, S., Yu, D., Huang, D., Qin, H., 2013. Cloud motion estimation for short term solar irradiation prediction. In: *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, pp. 696–701.
- Kazantzidis, A., Tzoumanikas, P., Bais, A., Fotopoulos, S., Economou, G., 2012. Cloud detection and classification with the use of whole-sky ground-based images. *Atmos. Res.* 113, 80–88.
- Lave, M., Kleissl, J., 2013. Cloud speed impact on solar variability scaling-application to the wavelet variability model. *Solar Energy* 91, 11–21.
- Leese, J.A., Novak, C.S., Taylor, V.R., 1970. The determination of cloud pattern motions from geosynchronous satellite image data. *Pattern Recogn.* 2 (4), 279–292.
- Leese, J.A., Novak, C.S., Clark, B.B., 1971. An automated technique for obtaining cloud motion from geosynchronous satellite data using cross correlation. *J. Appl. Meteorol.* 10 (1), 118–132.

- Lindeberg, T., 1993. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. *Int. J. Comput. Vis.* 11 (3), 283–318.
- Long, C.N., Sabbburg, J.M., Calbó, J., Pagès, D., 2006. Retrieving cloud characteristics from ground-based daytime color all-sky images. *J. Atmos. Ocean. Technol.* 23 (5), 633–652.
- Lucas, B.D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI'81*, vol. 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 674–679. URL <<http://dl.acm.org/citation.cfm?id=1623264.1623280>>.
- Marquez, R., Coimbra, C.F., 2013. Intra-hour dni forecasting based on cloud tracking image analysis. *Solar Energy* 91, 327–336.
- Marquez, R., Pedro, H.T., Coimbra, C.F., 2013. Hybrid solar forecasting method uses satellite imaging and ground telemetry as inputs to anns. *Solar Energy* 92, 176–188.
- Mori, N., Chang, K.-A., Introduction to mpiv, User Manual and Program. Available online at <<http://sauron.civil.eng.osaka-cu.ac.jp/~mori/software/mpiv>>.
- Nguyen, D.A., Kleissl, J., 2014. Stereographic methods for cloud base height determination using two sky imagers. *Solar Energy* 107, 495–509.
- Peng, Z., 2016. Multi-Source Image Integration Towards Solar Forecast, Ph.D. thesis. <<https://bitbucket.org/dtyu/solarpredictiondissertation/src/>>.
- Peng, Z., Yoo, S., Yu, D., Huang, D., 2013. Solar irradiance forecast system based on geostationary satellite. In: *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, pp. 708–713.
- Peng, Z., Yoo, S., Yu, D., Huang, D., Kalb, P., Heiser, J., 2014. 3d cloud detection and tracking for solar forecast using multiple sky imagers. In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, pp. 512–517.
- Peng, Z., Yu, D., Huang, D., Heiser, J., Yoo, S., Kalb, P., 2015. 3d cloud detection and tracking system for solar forecast using multiple sky imagers. *Solar Energy* 118 (0), 496–519. <http://dx.doi.org/10.1016/j.solener.2015.05.037>. URL <<http://www.sciencedirect.com/science/article/pii/S0038092X15002972>>.
- Pfister, G., McKenzie, R.L., Liley, J.B., Thomas, A., Forgan, B.W., Long, C.N., 2003. Cloud coverage based on all-sky imaging and its impact on surface solar irradiance. *J. Appl. Meteorol.* 42 (10), 1421–1434.
- Rutledge, G., Legg, E., Menzel, P., 1991. Operational production of winds from cloud motions. *Global Planet. Change* 4 (1), 141–150.
- Savoy, F.M., Lemaitre, J.C., Dev, S., Lee, Y.H., Winkler, S., 2015. Cloud base height estimation using high-resolution whole sky imagers. In: *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.
- Shields, J.E., Johnson, R.W., Koehler, T.L., 1993. Automated whole sky imaging systems for cloud field assessment. In: *Fourth Symposium on Global Change Studies*. American Meteorological Society, pp. 228–231.
- Shusterman, E., Feder, M., 1994. Image compression via improved quadtree decomposition algorithms. *IEEE Trans. Image Process.* 3 (2), 207–215.
- Singh, M., Glennen, M., 2005. Automated ground-based cloud recognition. *Pattern Anal. Appl.* 8 (3), 258–271.
- Sotiras, A., Davatzikos, C., Paragios, N., 2013. Deformable medical image registration: a survey. *IEEE Trans. Med. Imag.* 32 (7), 1153–1190.
- Stone, H.S., Orchard, M.T., Chang, E.-C., Martucci, S.A., 2001. A fast direct fourier-based algorithm for subpixel registration of images. *IEEE Trans. Geosci. Rem. Sens.* 39 (10), 2235–2243.
- Stuhlmann, R., Rieland, M., Paschke, E., 1990. An improvement of the IGMK model to derive total and diffuse solar radiation at the surface from satellite data. *J. Appl. Meteorol.* 29 (7), 586–603.
- Sun, D., Roth, S., Black, M.J., 2010. Secrets of optical flow estimation and their principles. In: *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2432–2439.
- Sun, D., Roth, S., Black, M.J., 2014. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *Int. J. Comput. Vis.* 106 (2), 115–137.
- Thirion, J.-P., 1998. Image matching as a diffusion process: an analogy with maxwell's demons. *Med. Image Anal.* 2 (3), 243–260.
- Urquhart, B., Kurtz, B., Dahlin, E., Ghonima, M., Shields, J.E., Kleissl, J., 2014. Development of a sky imaging system for short-term solar power forecasting. *Atmos. Meas. Tech. Discuss.* 7 (5), 4859–4907.
- Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D., 2009. An improved algorithm for tv-l 1 optical flow. In: *Statistical and Geometrical Approaches to Visual Motion Analysis*. Springer, pp. 23–45.
- West, S.R., Rowe, D., Sayeef, S., Berry, A., 2014. Short-term irradiance forecasting using skycams: motivation and development. *Solar Energy* 110, 188–207.
- Wood-Bradley, P., Zapata, J., Pye, J., 2012. Cloud tracking with optical flow for short-term solar forecasting. In: *50th Conference of the Australian Solar Energy Society*, Melbourne.
- Xu, J., Yoo, S., Yu, D., Huang, H., Huang, D., Heiser, J., Kalb, P., 2015. A stochastic framework for solar irradiance forecasting using condition random field. In: *Advances in Knowledge Discovery and Data Mining*. Springer, pp. 511–524.
- Xu, J., Yoo, S., Yu, D., Huang, D., Heiser, J., Kalb, P., 2015. Solar irradiance forecasting using multi-layer cloud tracking and numerical weather prediction. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC '15*. ACM, New York, NY, USA.
- Yang, H., Kurtz, B., Nguyen, D., Urquhart, B., Chow, C.W., Ghonima, M., Kleissl, J., 2014. Solar irradiance forecasting using a ground-based sky imager developed at UC san diego. *Solar Energy* 103, 502–524.
- Zhang, Q., Xu, L., Jia, J., 2014. 100+ times faster weighted median filter (wmf). In: *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2830–2837.